

35-05

# CORBA/SNMP ゲートウェイの実装と評価

吉原 貴仁      堀内 浩規      小花 貞夫

国際電信電話株式会社 研究所

## 1. はじめに

筆者らは、分散オブジェクトの業界標準である CORBA (Common Object Request Broker Architecture)<sup>[1]</sup> によるネットワーク管理システム構築の際重要となる TMN (電気通信管理網)<sup>[2]</sup> や SNMP (Simple Network Management Protocol)<sup>[3]</sup> のネットワーク管理プロトコルを持つ装置 (以下、それぞれ TNM 装置, SNMP 装置と呼ぶ) の CORBA 環境への収容方式<sup>[4]</sup> を提案している。本稿では、SNMP を対象とした収容方式を実現する CORBA/SNMP ゲートウェイ (以下、CORBA/SNMP GW と呼ぶ) の実装とその評価結果を述べる。

## 2. CORBA 環境への SNMP の収容方式<sup>[4]</sup>

(1) SNMP の管理情報と IDL (インタフェース定義言語) 定義の対応付け

X/Open と NMF (Network Management Forum) の JIDM (Joint Inter Domain Management)<sup>[5]</sup> の仕様に従い、グループまたはテーブルエンタリにインタフェースを、オブジェクトタイプに IDL 属性を、また ASN.1 データ型に IDL データ型を対応付ける。

(2) IDL 定義の拡張ならびに SNMP 管理操作と IDL 操作の対応付け

Get/SetRequest, GetResponse ならびに Trap は上記 (1) の IDL 属性を利用し、IDL 操作に対応付ける。また、新たなインタフェースの定義により、GetNextRequest を IDL 操作に、既存インタフェースの拡張により、VarBindList を使った複数の SNMP オブジェクトへの管理操作を単一の IDL 操作に対応付ける。

SNMP 装置に対して、障害検出や SNMP オブジェクトの動的な変化を検出するため、ポーリングを行なうが、この際、CORBA オブジェクトの生成や削除を行なう CORBA オブジェクト "Factory" のインタフェースを新たに定義する。

(3) オブジェクトの名前の対応付け

SNMP オブジェクトの名前から CORBA オブジェクトの名前を生成するための規則を設ける。

## 3. CORBA/SNMP GW の実装

### 3.1. 実装方針

- (1) 2. 節の収容方式に基づいて実装する。
- (2) SNMP の任意の MIB (Management Information Base) 定義に対応可能とするため、CORBA/SNMP GW プログラムを自動生成させる。
- (3) CORBA は IONA 社 Orbix, 計算機は SUN, プログラム開発言語は C++ を使用する。

### 3.2 システム構成

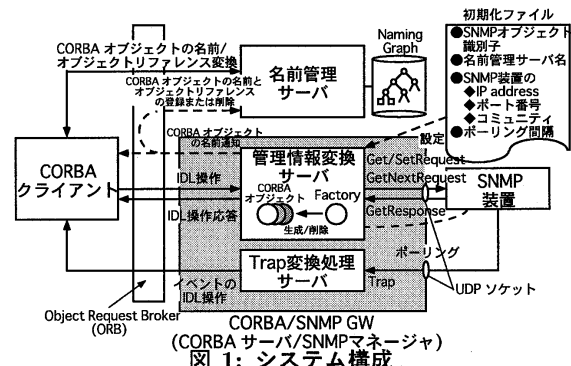
CORBA/SNMP GW は、1) 管理情報交換サーバ、および 2) Trap 変換処理サーバからなる (図 1)。ここでは、Trap は SNMP 装置から非同期に発行されるため、Trap の変換処理と他の管理操作の変換処理を別のサーバとして実現する。

両サーバは 2. 節 (1) と (2) に基づき、SNMP の管理

情報と IDL 定義との変換、ならびに SNMP の管理操作と IDL 操作との変換を行なう。

さらに、管理情報交換サーバは CORBA/SNMP GW 起動時または CORBA オブジェクトが生成される時、CORBA オブジェクトの名前とオブジェクトリファレンスの組みを名前管理サーバに登録する。

また、SNMP オブジェクト識別子、名前管理サーバ名、SNMP 装置の IP address, ポート番号, コミュニティ等の動作環境、ならびにポーリング間隔をユーザが独自に初期化ファイルに設定し、管理情報交換サーバが起動時に読み込む。

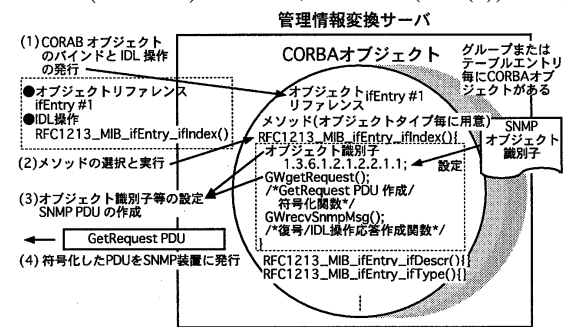


### 3.3. CORBA/SNMP GW の処理

(1) 管理情報交換サーバの処理

#### ● IDL 操作の処理

CORBA クライアントはオブジェクトリファレンスにより、SNMP のグループまたはテーブルエンタリに対応する CORBA オブジェクトにバインドし、オブジェクトタイプの値取得または設定のための IDL 操作を発行する (図 2(1))。CORBA オブジェクトは、IDL 操作の名前をキーとして、オブジェクトタイプ毎に用意された関数 (メソッド) を選択し、実行する (図 2(2))。



このメソッドはすべての CORBA オブジェクトに共通の SNMP PDU 作成ライブラリ (例えば、GWgetRequest()) を用いて、オブジェクト識別子等のパラメータを設定し (図 2(3)), また、符号化した PDU を SNMP 装置に発行する (図 2(4))。

GetRequest に対する GetResponse を受信した場合、

取得値を SNMP PDU 作成ライブラリで復号し、IDL 操作応答に詰め替えて CORBA クライアントに返す。

GetNextRequest および複数 SNMP オブジェクトへの管理操作に対する GetResponse を受信した場合、取得値の SNMP オブジェクトを特定し、対応する CORBA オブジェクトの名前を CORBA クライアントに返答する必要がある。このため、GetResponse の VarBindList に含まれる SNMP オブジェクト識別子から、対応するオブジェクトリファレンスを抽出し、名前管理サーバで CORBA オブジェクトの名前に変換させ、取得値と合わせて CORBA クライアントへ返答する。

#### ● SNMP オブジェクトの生成/削除の検出

管理情報変換サーバは、予め設定されたポーリング間隔で、SNMP 装置へ GetNextRequest を順次発行し、装置にあるグループまたはテーブルエントリのオブジェクトを取得し、オブジェクトの生成/削除を検出する。

新たなオブジェクトがある場合、オブジェクトを生成する IDL 操作を CORBA オブジェクト “Factory” に発行する。“Factory” は CORBA オブジェクトを生成し、名前管理サーバに CORBA オブジェクト (名前とオブジェクトリファレンス) の登録を依頼する。

一方、オブジェクトが削除された場合、対応する CORBA オブジェクトを削除する IDL 操作を “Factory” に発行し、また、名前管理サーバに当該 CORBA オブジェクトの登録抹消を依頼する。

#### (2) Trap 変換処理サーバの処理

Trap を受信すると、これに含まれる IP アドレス、enterprise オブジェクト識別子、および generic-trap 番号等の値からイベントの IDL 操作を作成し、CORBA クライアントへ通知する。

### 3.4. プログラムの自動生成

SNMP の MIB 定義を入力とし、CORBA/SNMP GW プログラムを C++ 言語で自動生成する CORBA/SNMP GW プログラムジェネレータ (図 3) を実装した。

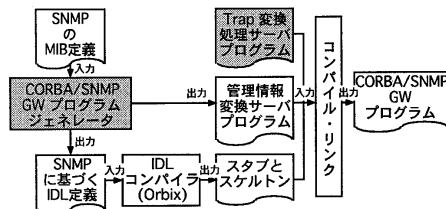


図 3: プログラムの自動生成

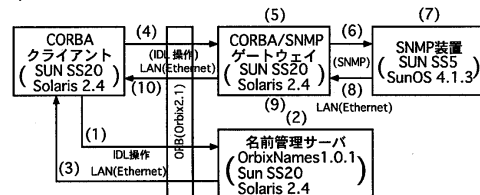
IDL 操作の仲介をする ORB と CORBA オブジェクト間のソフトウェアインタフェースを含むスタブ、および CORBA オブジェクトのクラス定義やメソッドの雛型等を含むスケルトンを Orbix の IDL コンパイラにより生成させる。また、SNMP のオブジェクト識別子や SNMP PDU 作成ライブラリ等の各メソッドの実装 (図 2 の CORBA オブジェクトの点線枠内) を管理情報変換サーバプログラムとしてプログラムジェネレータに生成させる。Trap は SNMP のグループやテーブルエントリの種類に依存しないため、共通の Trap 変換処理サーバプログラムを作成した。上記 3 つをコンパイル・リンクし、CORBA/SNMP GW プログラムを生成する。

RFC1213<sup>[6]</sup> (MIB-II) に基づく (17 個のグループまたはテーブルエントリを含む) 2.6 Kstep の MIB 定義の場合、図 3 の Trap 変換処理サーバプログラムは 0.2 Kstep、管理情報変換サーバプログラムは 9.4Kstep、スタブは 7.8Kstep、スケルトンは 6.7 Kstep であった。

### 4. プログラムの処理時間

RFC1213<sup>[6]</sup> (MIB-II) の MIB 定義に対して、自動生成した CORBA/SNMP GW プログラムを使用し、処理時間を計測した。図 4 に試験構成と測定点を、表 1 に処理時間をそれぞれ示す。

SNMP 装置の処理時間 (図 4 (6) から (8) の和) が 1.5msec から 3.0msec 程度であることを考えると、表 1 のいずれの測定項目においても CORBA/SNMP GW のオーバーヘッドは小さい。また、表 1 (e) は、CORBA/SNMP GW での値の詰め替えが他の SNMP の管理操作と比較して数多く生じるため、処理時間が長い。



- (1)と(3)はCORBAクライアントと名前管理サーバ間の通信時間
- (4)と(10)はCORBAクライアントとCORBA/SNMPゲートウェイ間の通信時間
- (6)と(8)はCORBA/SNMPゲートウェイとSNMP装置間の通信時間
- (2)は名前管理サーバの処理時間
- (5)と(9)はCORBA/SNMPゲートウェイの処理時間
- (7)はSNMP装置の処理時間

図 4: 試験構成と測定点  
表 1: プログラムの処理時間

測定項目	処理時間 (msec)
(a) RFC1213_MIB_ifEntry_ifIndex() による ifIndex (INTEGER 型) の値取得	6.6 7.2 23.1
(b) RFC1213_MIB_system_sysDescr() による sysDescr (DisplayString 型) の値取得	7.6 7.0 26.3
(c) RFC1213_MIB_system_sysLocation() による sysLocation (DisplayString 型) の値設定	6.2 6.9 18.7
(d) get_next() による ifDescr (DisplayString 型) の値取得	10.4 7.2 28.4
(e) snmpTrap() による Trap (coldStart) の受信	9.5 13.0 -

- (注 1) 上段の数値は図 4 (5) と (9) の和、中段の数値は図 4 (1)、(2)、および (3) の和、下段の数値は図 4 (1) から (10) までの和。
- (注 2) (e) の上段の数値は図 4 (9)、(e) の中段の数値は (8)、(9)、および (10) の和。
- (注 3) 各数値は 100 回の試行の平均値。

### 5. おわりに

分散オブジェクト (CORBA) 環境への SNMP 装置収容のための CORBA/SNMP GW を実装し、処理時間を評価した。実装では、任意の MIB 定義に対して、プログラムの自動生成が有効である。最後に日頃御指導頂く KDD 研究所村上所長に感謝する。

### 参考文献

- [1] Object Management Group. *The Common Object Request Broker: Architecture and Specification Revision 2.0*, Jul. 1995.
- [2] CCITT. *Recommendation M.3010, Principles for Telecommunications Management Network*, 1992.
- [3] Case J. et al. *A Simple Network Management Protocol (SNMP)*. IETF, RFC 1157, May 1990.
- [4] 堀内 吉原 小花. 分散オブジェクトによるネットワーク管理のための TMN/SNMP 収容方式. 情報第 55 回全大, 3S-04, Sep 1997.
- [5] X/Open Preliminary Specification. *Inter-Domain Management Specifications: Specification Translation*, Nov. 1996.
- [6] K. McCloghrie et al. *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. IETF, RFC 1213, Mar. 1991.